

מדינת ישראל  
STATE OF ISRAEL

Ministry of Justice  
Patent Office

משרד המשפטים  
לשכת הפטנטים

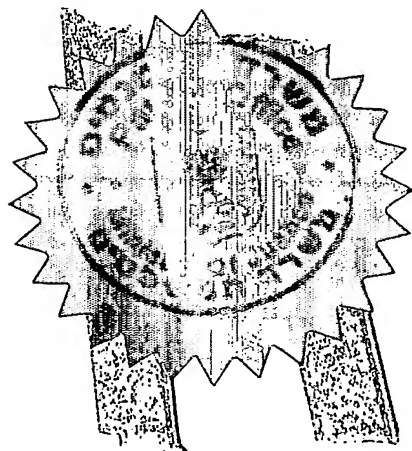
IB/04/3228

This is to certify that annexed  
hereto is a true copy of the  
documents as originally  
deposited with the patent  
application of which  
particulars are specified on the  
first page of the annex.

זאת לתעודה כי רצופים בזה  
העתקים נכונים של המסמכים  
שהופקדו לכתחילה עם  
הבקשה לפטנט לפי הפרטים  
הרשומים בעמוד הראשון של

הנספח.

**PRIORITY  
DOCUMENT**  
SUBMITTED OR TRANSMITTED IN  
COMPLIANCE WITH RULE 17.1(a) OR (b)



16-11-2004  
This is the true and correct copy of the original document submitted to the  
מ.ל.ס.  
ממשרד הפטנטים  
Commissioner of Patents

נתאשר  
Certified

BEST AVAILABLE COPY

# בקשה לפטנט

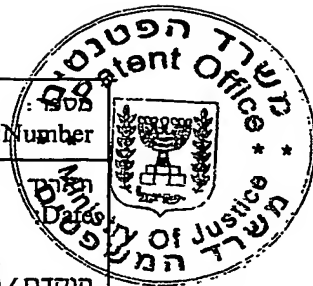
Application for Patent

158282

Number

03-10-2003

Ante / Post-dated



אני, (שם המבקש, מענו - ולגבי גוף מאוגד - מקום התאגדותו)

I (Name and address of applicant, and, in case of body corporate place of incorporation)

Netmask (El-Mar) Internet Technologies Ltd.  
8 Yad Harutzim Street  
Kfar Saba 44641  
Israel

נטמאסק (אל-מר) טכנולוגיות אינטרנט בע"מ  
רח' יד חרוצים 8  
כפר סבא 44641  
ישראל

שמה הוא \_\_\_\_\_ Law  
Of an invention, the title of which is

בעל אמצאה מכח \_\_\_\_\_ הדין  
Owner, by virtue of

קביעת תצורה (בעברית)  
(Hebrew)

Configuration Setting

(באנגלית)  
(English)

Hereby apply for a patent to be granted to me in respect thereof

מבקש בזאת כי ינתן לי עליה פטנט

*בקשת חלוקה - Application of Division		*בקשת פטנט מוסף - Application for Patent Addition		*דרישה דין קדימה Priority Claim		
מבקשת פטנט from Application	לבקשה/לפטנט to Patent/Appl.	מספר/ סימן Number/Mark	תאריך Date	מדינת האגוד Convention Country		
No. _____ מסי	No. _____ מסי					
Dated _____ מיום	Dated _____ מיום					
*יפוי כח כללי/מיוחד (רצוף בזה) / עוד יוגש P.O.A: general / individual - attached / to be filed later הוגש בעניין _____ 129633 filed in case						
המען למסירת הודעות ומסמכים בישראל פוסטאר ושות' פניין רוחני 2002 בע"מ רח' בזל 16 פ"ת ת.ד. 10256 פ"ת, 49002						
עבור המבקש, חתימת המבקש Signature of Applicant		שנת 2003 Of the year		בחודש אוקטובר Of		
פוסטאר ושות' פניין רוחני 2002 בע"מ				היום 2 This		
				לשימוש חלשכה For Office Use		

018/03747

טופס זה, כשהוא מוטבע בחותם לשכת הפטנטים ומושלם מספר ובתאריך ההגשה, הינו אישור להגשת הבקשה שפרטיה רשומים לעיל.

This form, impressed with the Seal of the Patent Office and indicating the number and date of filing, certifies the filing of the application, the particulars of which are set out above.

קביעת תצורה

Configuration Setting

אליהו מרמור

Eliyahu MARMOR

## קביעת תצורה

## References Cited:

1. <http://msdn.microsoft.com/library/default.asp?url=/workshop/browser/editing/mshtmleditor.asp>
2. <http://www.mozilla.org/editor/midas-spec.html>
3. <http://www.microsoft.com/catalog/display.asp?subid=22&site=732&pg=1>
4. <http://www.ietf.org/rfc/rfc2616.txt>
5. [http://www.google.com/machine\\_translation.html](http://www.google.com/machine_translation.html)
6. <http://babel.altavista.com/>
7. <http://www.wiztom.com/>
8. US 6,601,108 Marmor

## רקע

כשה-web הומצא, בתחילת שנות התשעים, הכוונה היתה ליצור רשת ומימשקים לגישה למידע. הממציאים לא חלמו שהמצאתם תכבוש את העולם, ואף תהפוך למימשק פופולרי להפעלת תוכנות. למרות שהדפדפנים היו מאוד פרימיטיביים יחסית להיום, הם נבדלו מאוד זה מזה, ורמת התאימות ביניהם היתה נמוכה מאוד.

עם ההצלחה, באו גם דרישות הולכות וגוברות מצד המשתמשים, וגם רצון עז של יצרני הדפדפנים לכבוש את השוק. שני גורמים אלו דחפו את יצרני הדפדפנים להוסיף עוד ועוד פונקציונליות, עד כדי כך שהם היום גדולים באלפי אחוזים ממה שהיו בשנים הראשונות של ה-web. למרות התוספות הרבות, רמת התאימות בין הדפדפנים העיקריים (ובמיוחד Microsoft Internet Explorer ו-Mozilla) השתפרה.

אחת מהתכונות שנתמכת כיום ע"י הדפדפנים המובילים, היא היכולת להפעיל client-side-active-code ששוגרו אליו כחלק מהדף או בצמוד לו. רוב הדפדפנים תומכים הן ב-JavaScript, הן ב-Java, והן ב-Flash. קיימים הבדלים במימוש, אך הם קטנים למדי.

תכונה נוספת שנתמכת ע"י הדפדפנים המובילים, היא ההתייחסות אל המסמך כאל עץ; בין אם מדובר במסמך HTML ובין אם מדובר במסמך XML, יצרן הדף יכול להוסיף קוד (למשל script של JavaScript) שייגש אל חלקי המסמך כאל ענפים בעץ.

Client-side-active-code (כמו למשל JavaScript) יכול גם לבחון את הערכים הנוכחיים של הענפים השונים, את ה-attributes שלהם, ואף לשנות אותם באופן דינמי.

תכונה אחרונה שראויה לציון בטרם נמשיך הלאה, היא היכולת להגדיר חלקים מהמסמך (או את כל המסמך) כ-editable, הן בהגדרתם המקורית, והן דינמית ע"י client-side-active-code. הגדרה זו גורמת לאותם חלקים להתנהג כלפי המשתמש כאילו הם חלקי מסמך שנטען ע"י editor (כמו למשל מעבד תמלילים) ולא ע"י דפדפן. המשתמש יכול לבצע עליהם פעולות שונות, כמו למשל resize או שינוי הטקסט שבהם, במקום הפונקציונליות המקורית שלהם (למשל מעבר לדף חדש בעקבות לחיצה על hyperlink). עם זאת, התכונה הזאת לכשעצמה לא משנה את המסמך עצמו, אלא רק את אופן הצגתו למשתמש; יתירה מכך, השרת (או האתר) כלל לא מודע לכך שהמשתמש שינה משהו. תכונה זו הופיעה לראשונה ב-Internet-Explorer (במסגרת ה-MSHTML-Editing-Component<sup>[1]</sup>), והוסבה לדפדפנים מתחרים (כמו למשל Mozilla במסגרת פרויקט MIDAS<sup>[2]</sup>).

עד כה, שימשה תכונה זו לקלט של Rich-Text מהמשתמש. למשל, אתרי דואר אלקטרוני מבוסס דפדפן (web-mail), היו מוגבלים בעבר לכתיבה של e-mails פשוטים בלבד, שכללו pure-text. הם הפכו במהירות למיושנים יחסית ל-mailers הרגילים (למשל Outlook) שאפשרו לכתוב מסרים מורכבים, עם עיצוב, הדגשות, צבעים, hyper-links, פונטים, תמונות וכו'. התמיכה של הדפדפנים ב-editing אפשרה ל-web-mails לסגור את הפער, ואתרי web-mail מודרניים<sup>[3]</sup> לא נופלים מכלי ה-e-mail המתקדמים ביותר. דבר זה הושג ע"י החלפת אלמנט ה-textarea של HTML (שאליו הוזן קלט המשתמש בעבר) בשטח ריק (בדרך כלל אלמנט ה-DIV או IFRAME של HTML), שהוגדר כ-editable. סדר

הפעולות הוא: תוכנת ה-web-mail מייצרת דף שבו יש למשתמש אפשרות לכתוב e-mail, במקום textarea היא מייצרת שטח ריק המוגדר כ-editable, מוסיפה לחצן "Send", ו-JavaScript-callback של שתופס את event הלחיצה על ה-Send, ומבצע את הפעולה הבאה: הוא קורא את תוכן ה-HTML של מה שהצטבר בשטח הריק, ומשגר אותו (אחרי נטרול תוים מיוחדים שעלולים לשמש כפרצות בטיחות) בחזרה לשרת, שמשגר אותו כ-e-mail לנמען.

שימוש בתכונה זו עבור קליטת Rich-Text, מועיל במקרה נוסף: כלים לבניית אתרים. גם כלים אלו מייצרים שטח ריק שמוגדר כ-editable. סדר הפעולות גם כאן זהה לחלוטין ל-web-mail, מלבד סוף התהליך: במקום שהתוצאה תישלח כ-e-mail לנמען, היא נשמרת כקובץ HTML על הדיסק או כמסמך/template ב-web-application-server או כרשומה בבסיס-נתונים כלשהו.

בנוסף לתכונות הדפדפנים, יש כלי נוסף שנחוץ להכיר לשם הבנת הפטנט: Proxy.

<sup>[4]</sup> HTTP-proxy הינו כלי העומד בין אתרי web או תוכנות בעלות מימשק web, לבין דפדפנים. כלי זה מתחזה ל-client כלפי האתרים ו/או התוכנות, ול-server כלפי הדפדפנים, ובעצם הוא צינור ביניהם.

- במקור, תשתית ה-HTTP-Proxy הומצאה לשתי מטרות עיקריות:
1. caching של תכנים (אגירת דפים הנטענים מהרשת שוב ושוב, כך שבפעמים הבאות ייחסך זמן ההמתנה לטעינה כי הדפים יובאו מה-proxy ישירות, ולא מהרשת)
  2. גישור בין רשתות בעלות מרחבי כתובות שונים, או במילים אחרות – מתן אפשרות לגולשים בארגון המצויים מאחורי חומות firewall, לגשת לתכנים הנמצאים ברשת העולמית.

בהמשך, הופיעו מוצרים שלקחו את תשתית ה-proxy צעד אחד הלאה, ואפשרו ביצוע transformations לתכנים שעוברים דרך ה-proxy.

#### תאור הפטנט

פטנט זה יתאר שיטות חדשניות המאפשרות לבצע את הגדרת ה-transformation וה-customization של transformation-proxy במצב WYSIWYG. החדשנות באה לידי ביטוי גם בכך שהיא לא דורשת פיתוח של אף כלי עריכה, אלא רותמת את תכונות ה-editing הפנימיות של הדפדפנים לצורך הגדרת ה-transformations וה-customizations של transformation proxies, ובכלל זה גם הגדרות של שינויים שיש לבצע להבא בגוף המסמכים לפני שליחתם לדפדפנים, ולא רק של התוצאה הסופית של השינויים. היכולת לבצע שינויים ולשמור את השינויים עצמם ולא את התוצאה שלהם, מאפשרת לתמוך גם בתכנים דינמיים. כל השיטות הן בלתי-פולשניות ("non-intrusive"), ולא כרוכות בשום שינוי באף אחד מהצדדים, לא באתר, ולא בדפדפן (למעט – כאשר מדובר ב-forward-proxy ולא ב-reverse-proxy או ב-transparent-proxy – הגדרה של ה-proxy בהגדרות הדפדפן). בכל מקרה, לא צריך לשנות את תכני האתר, וגם לא צריך להתקין שום דבר בצד ה-client (מלבד דפדפן סטנדרטי כדוגמת Internet-Explorer או Mozilla), ודאי שלא plug-in כלשהו או client-application. לא נדרש אפילו אישור להתקנה אוטומטית של ActiveX או active-content כלשהו שדורש אישור משתמש לצורך התקנתו.

הרעיון מתבסס על כך שה-proxy שולח אל משתמשים מורשים מסמכים שעברו שיפוץ ע"י ה-proxy. שיפוץ זה גורם למסמך להיות editable, או מאפשר למשתמש לעבור למצב editing ע"י לחיצה על לחצן שה-proxy הוסיף לדף, או כלשהו אחר. בנוסף לכך, ה-proxy מוסיף לחצן שניתן ללחוץ עליו בתום השינויים לצורך שמירה, או trigger כלשהו אחר. הלחצן עשוי להיות כמובן אותו לחצן כניסה למצב ה-editing, שהתפקיד שלו התחלף עם הכניסה למצב editing, ומאותו רגע הוא משמש לסיום ה-editing. בנוסף לכל האמור עד כה, ה-proxy מוסיף למסמך גם שני קטעי client-side-active-code, אחד שמופעל לפני תחילת ה-editing (להלן: קטע ראשון), ואחד שמופעל בעת השלמת ה-editing (להלן: קטע שני). קטעים אלו עשויים להיות callbacks של הלחצנים הנ"ל, ובאופן ספציפי יותר – קטעי JavaScript. כדי לאפשר לקטע השני לגלות מה שונה ע"י השוואה, הקטע הראשון עשוי לרצות להעתיק את כל הנתונים הדרושים על הדף ועל תוכנו לפני שהוא הופך אותו ל-editable. העברת הדף

למצב editable עשויה להתבצע באופן סלקטיבי, ותוך דילוג על אלמנטים מסוימים. סלקציה זו שימושית בעיקר כדי לא להפוך את האלמנטים שה-proxy הוסיף למסמך, ל-editable בעצמם. כדי להקל על ביצוע סלקציה זו, ה-proxy עשוי לסמן את אותם אלמנטים, למשל ע"י מתן שמות ייחודיים, או הוספת attribute שיבדל אותם, או ע"י הכללתם מתחת ל-parent-node משותף שהקטע הראשון יידע לזהות.

הקטע השני, שמופעל בתום השינויים, עשוי לשגר את התוצאה הסופית ישירות ל-proxy, אך בדרך כלל נעדיף שהוא ישגר את השינויים עצמם, או יאפשר לנו לעבד אותם ולהוסיף להם כללים והגדרות, למשל באמצעות dialog box. דרך יעילה לאתר את השינויים, היא רקורסיה או השוואה היררכית כלשהי, שסורקת את עץ האלמנטים של הדף, בין אם הוא דף HTML, XML, או מה שלא יהיה, ומשווה את הערך הנוכחי לערך שנשמר ע"י הקטע הראשון. בתום איתור השינויים, הקטע יכול לשגר את הפרטים השונים ישירות ל-proxy, או לחילופין לפתוח dialog-box שיאפשר לשנות ו/או להוסיף הגדרות ל-transformation-rule לפני שיגורו ל-proxy.

שימוש מועיל ביותר לפטנט, יכול להיות במקרה שהמשתמש המורשה מעוניין להחליף מילים בגוף המסמך, ע"י מילים אחרות, למשל לצורך תרגום סטאטי (השימוש במילה "סטאטי" נועד כדי להבדיל מ-Machine-Translation, אך כמובן שתרגום זה מיושם גם על מסמכים דינמיים). משתמש זה עשוי לרצות להשתמש ב-translation-proxy כדוגמת <sup>[7]</sup>WizTom, ובניגוד למצב הנוכחי, הוא יוכל לבצע את התרגומים במימשק WYSIWYG (What You See Is What You Get) מלא, בגוף הדפדפן, וללא טעינה של שום תוכנה אחרת.

במקרה שמדובר בהחלפת טקסט בטקסט, ה-dialog-box שייפתח עשוי לכלול שדות קלט המכילים את הטקסט המקורי ואת הטקסט המחליף אותו. המשתמש יכול לשנות את התוכן המדויק של טקסטים אלו לפני השיגור ל-proxy. ה-transformation יתבצע בכל מקרה שיימצא match מדויק של הטקסט המקורי המופיע ב-dialog-box זה. מימושים מתוחכמים יותר של הפטנט יאפשרו למשתמש להגדיר שה-match לא חייב להיות מדויק, אלא יכול להתבסס על regular-expression, או על case-insensitive-match, או כל approximate-match אחר על פי בחירת המשתמש ב-dialog-box זה.

בשדה הקלט של הטקסט החדש, המשתמש יוכל באופן אופציונלי לבחור אחת מאפשרויות קיימות, במקום להקליד ידנית תוכן ו/או במקום לשנות ידנית את התוכן הקיים. האפשרויות הקיימות יוצגו לו בצורת selection-box או pulldown או pop-up או כל אמצעי אחר שמאפשר בחירה בין כמה אפשרויות, והן תילקחנה מהפלט של אלגוריתם שיופעל על הטקסט המקורי. האלגוריתם עשוי להיות מילון או מכונה המתרגמת את הטקסט המקורי משפת מקור מסוימת לשפת יעד מסוימת, או Translation-Memory המציג בפני המשתמש את הטקסטים הקודמים שהוזנו ע"י המשתמשים השונים כתחליף לטקסט מקורי זהה לנוכחי.

מנגנון זה, מאפשר למשתמשים לתרגם מסמכים בלי צורך להקליד בכלל; הם יכולים לעבור למצב editable ע"י הלחצן המתאים, לסמן מילה, למחוק אותה (כלומר להחליף אותה באורך 0 תווים), וללחוץ על הלחצן שפותח את ה-dialog-box. בחלון שייפתח להם, יופיע הטקסט המקורי, ובמקום הטקסט החדש יופיע שדה ריק. בעזרת מנגנון הבחירה, יוכלו המשתמשים לבחור את התרגום המתאים ביותר של המילה המקורית מתוך שלל התרגומים המוצגים בפניהם. לחילופין, הם יכולים לעיין בתרגומים השונים שהמשתמשים הזינו עבור אותו טקסט מקורי, ולבחור באחד מהם. כל מה שנותר להם עתה הוא לסיים את העבודה ב-dialog-box (ע"י לחיצה על לחצן מתאים, או באופן אחר כלשהו).

אם הקוד שמייצר את ה-dialog-box נשלח לדפדפן כחלק מהמסמך המקורי, הוא לא יכול לכלול את התרגומים השונים או את ה-Translation-Memory, משום שהקוד נוצר עוד לפני שהמשתמש ביצע את השינויים במסמך. כתוצאה מכך, ה-dialog-box עלול להיאלץ לשלוח את הטקסט המקורי ל-proxy, ולקבל ממנו כתשובה את האפשרויות השונות לתרגום. שימוש ב-HTTP-request רגיל לצורך כך הוא בעייתי, משום שהוא גורם להחלפת התוכן של ה-dialog-box בתוכן חדש. ה-proxy יכול אמנם לשלוח תוכן החדש שיהיה זהה לתוכן הרצוי של ה-dialog-box (בתוספת אפשרויות התרגום השונות ו/או ה-

Translation-Memory עבור הטקסט המקורי), אך תהליך זה עדיין איננו שקוף לחלוטין למשתמש, ולכן אנחנו עשויים לרצות לבצע תהליך זה באופן הבא:

1. בנוסף ל-dialog-box, נוצר frame נוסף נסתר (size=0, border=0)
2. ה-hidden-frame משגר אוטומטית את הטקסט המקורי ל-proxy ע"י HTTP-request
3. ה-proxy משגר כ-response דף שמוגדר שמיד עם טעינתו בדפדפן הוא יגש לשדה ב-dialog-box בו מוצגים התרגומים השונים למשתמש, ויציב בשדה זה את הערכים המתאימים שהתקבלו מה-proxy. פעולה זו תבוצע ע"י client-side-active-code, למשל JavaScript

כמובן, הפטנט שימושי לא רק עבור החלפת טקסט סטטית, אלא עבור כל שינוי במסמך, החל משינוי עיצוב (למשל הדגשה של מילה), דרך שינוי כיוון המסמך (ראה [8] Netmask) ועוד אינספור transformations שונים ומשונים, וכלה בהגדרת אזורים שלמים בגוף המסמך כממעדים ל-transformation שעתיד להתבצע עליהם בכל מעבר שלהם דרך ה-proxy, למשל עבור תרגום פונטי שלהם (תרגום שם לפי צליל ובלי לשמור על המשמעות המקורית), או Machine-Translation של כל אזור בגוף המסמך לשפה משלו. נניח למשל שברצוננו לתרגם חשבונית, בעלת כותרות ותוויות טקסט קבועות ("Name:", "Price:", "VAT:" וכו'), וכן ערכים דינמיים שכוללים מספרים (שלגביהם התרגום לא רלבנטי) ושמות ("John", "David" וכו'), שאותם יש לתרגם פונטית). נניח שה-proxy שלנו מאפשר הן Machine-Translation, והן תרגום פונטי של השמות. קיימים כיום כלים שמבצעים כל אחת מהמשימות האלו בנפרד. למשל, [5] google ו-[6] Babel-Fish כוללים proxy שמסוגל לבצע Machine-Translation על תכנים שעוברים דרכו. הבעיה היא שאם החשבונית תכלול את המילה "bill" במשמעות של חיוב, וכן את השם "bill" כציון לאדם מסוים, Machine-Translation עלול לתרגם את שניהם כ"חיוב", ותרגום פונטי עלול לתרגם את שניהם כאילו היו שמות. משום כך, רצוי שה-dialog-box יכולת אפשרות להגביל כל transformation לקונטקסט מסוים בדף.

כדי לעשות זאת, ה-dialog-box עשוי לכלול שדה קלט שמתאר קונטקסט שכזה. שפה יעילה ביותר לתיאור קונטקסט בדף web, הינה XPath, הכלולה בשפת XSLT. השפה הומצאה לצורך טרנספורמציות על XML, אך ניתן להשתמש בה גם ל-HTML. היות ורוב המשתמשים יתקשו לייצר ביטויים בשפה זו, שדה הקלט יקבל אוטומטית ערך התחלתי שיתייחס לקונטקסט בו בוצע השינוי המסוים הזה, כך שכל שנותר למשתמש לעשות הוא לסמן או לבטל את ה-checkbox שליד שדה הקלט, ההופך את ההגדרה הזו לפעילה, או לבטל סימון קיים. משתמש חכם יוכל לשנות את הערך ההתחלתי, כך שיהיה יותר כללי או פחות כללי. כדי להקל על המשתמש, אפשר גם להוסיף לחצן שלחיצה עליו תסמן או תדגיש את חלקי המסמך שהגדרת ה-XPath הנוכחית כוללת. לחצן דומה יכול "לסמל" את ביצוע ה-transformation, או ביטולו, על המסמך, ולהראות באופן מידי מה היה קורה אילו המסמך היה עובר שינוי ע"י ה-proxy על סמך transformation rule זה.

שדות קלט נוספים שעשויים להיכלל ב-dialog-box הם שדות קלט שמגבילים את ביצוע ה-transformation לא לקונטקסט מסוים בדף, אלא גם לדפים מסוימים. שדה קלט אחד שכזה, יכיל את הכתובת של הדף, או חלקים ממנה (למשל ה-hostname, ה-port, וכו'). ערכו ההתחלתי עשוי להילקח מהכתובת של הדף אותו שינה המשתמש, והמשתמש יוכל לשנות את ערך השדה הזה, ו/או להדליק/לכבות את ה-checkbox שליד שדה זה שמציין אם אכן להגביל את ה-transformation לדף המוגדר. מימוש מתוחכם יותר של הפטנט עשוי לאפשר למשתמש להגדיר שערך השדה הזה הוא regular expression ולא בהכרח ערך מדויק. שדות קלט אופציונליים נוספים עשויים להכיל את שם ה-authenticated-user שרק עבור requests שלו יבוצע ה-transformation, ערכי פרמטרים של ה-request שגרם ליצירת הדף (כמו למשל HTTP-parameters או HTTP-parameters), ערכי ה-HTTP cookies, ובעצם כל נתון שכלול ב-HTTP-headers של ה-request או בגוף ה-request. dialog-box לדוגמה מופיע בשרטוט מספר 1.

קיימות טרנספורמציות נוספות ש-transformation-proxies נודגים לבצע, ושהפטנט המוצע מאפשר להגדיר אותן בממשק web, ברמת WYSIWYG, תוך רתימת יכולות ה-editing הפנימיות של

הדפדפנים: אחת הטרנספורמציות המתוארות ב-[8] הופכת את הכיוון הראשי של המסמך (או חלקים ממנו), עבור שפות כמו עברית וערבית. בזכות התמיכה של יכולות ה-editing הפנימיות של הדפדפנים בשינוי הכיוונים, טרנספורמציה שכזאת יכולה להיות מוגדרת ע"י כניסה למצב editing, סימון הקטע המיועד לשינוי כיוון, ביצוע השינוי בגוף הדפדפן (למשל ע"י לחצן כיוונים), ושמירה (או כניסה ל-dialog-box). השיטה המתוארת לעיל לגבי השוואת המסמך לפני ואחרי ה-editing, תאפשר לאתר את השינוי במהירות, וליצור transformation rule מתאים.

אולם הפטנט מאפשר להגדיר אזורים במסמך גם בלי שהמשתמש ביצע עליהם שינוי שאותר ע"י ההשוואה הנ"ל. דבר זה מתבצע ע"י היכולת של client-side-active-code לזהות אלו אזורים במסך מסומנים, ולפתוח dialog box (או לשגר ישר transformation rule עבור אזורים אלו) מיד עם סיום ה-editing (שכאמור, במהלכו לא שונה דבר, אלא רק סומנו אזורים במסך).

יכולת זאת מאפשרת הגדרות כמו "תרגום קטעים אלו תרגום פונטי" או "תרגום קטעים אלו ע"י Machine-Translation", שתוארו לעיל.

לעתים יש להעמיד לרשות המשתמש תוכנת עריכה מיוחדת לאחד האלמנטים הכלולים בדף. הסתמכות על תוכנה מותקנת בצד המשתמש סותרת את כלל ה-non-intrusive שהצבנו, מכמה בחינות:

1. תלות בתוכנה שעל המשתמש להתקין
2. לא ידוע אם התוכנה הנ"ל מסוגלת בכלל לרוץ על פלטפורמת המחשב של המשתמש
3. המשתמש חייב לשמור את האלמנט כקובץ במחשב שלו
4. המשתמש חייב להפעיל ידנית את התוכנה
5. המשתמש צריך לשגר את התוצאה בחזרה ל-HTTP-intermediary

כדי לפתור בעיה זו, הפטנט מציע להשתמש בשילוב של remote-display ושל viewer הכתוב כ-client-side-active-code. שילוב זה יאפשר להפעיל את התוכנה בצד של ה-HTTP-intermediary (באותו מחשב או על מחשב המקורש אליו), ועדיין לאפשר למשתמש להשתמש בתוכנה כאילו רצה בתוך הדפדפן שלו. הדבר מושג באמצעות display וירטואלי שה-HTTP-intermediary מייצר, כדוגמת VNC או RDP, הרצת התוכנה עליו, ושיגור viewer שמסוגל לשמש כ-client של ה-display הוירטואלי אל הדפדפן. ה-viewer יכתב בשפה שתאפשר לו לרוץ אוטומטית על הדפדפן, ללא כל צורך בהתקנת plug-in כלשהו או אישור כלשהו מצד המשתמש. בדרך כלל יהיה מדובר ב-client-side-active-code, כמו למשל Java.

ה-viewer יציג בתוך הדפדפן את תוכנת העריכה, למרות שהיא רצה בכלל במחשב אחר, ויאפשר למשתמש להפעיל אותה, ולצאת בתום העריכה, בין אם בלויית save ובין אם לאו.

ה-viewer עשוי לכלול תכונות רבות שאינן רלבנטיות ל-remote-display שכזה, ושיבלבלו את המשתמש. כמו כן, תכונות מסוימות עלולות להיות מסוכנות לאבטחה, ולשמש כפרצות. למשל, המשתמש יכול להשתמש ב-dialog-box של File->Open כדי לחקור את תוכן הדיסק שעליו רצה תוכנת העריכה. באמצעות File->Save הוא עלול אף לשנות את תוכן הדיסק. הוא יכול להשתמש גם בתפריטים ובמישק של ה-window-manager של ה-display הוירטואלי, כדי לבצע את זממו. לכן, הסתעפויות מסוימות של הפטנט יאפשרו למנוע סכנות אלו ע"י סירוס חלק מהאופציות של ממשק המשתמש של תוכנת העריכה ושל ה-window-manager, או אף נטרול ה-window-manager לחלוטין.



## Claims

1. שימוש ב-HTTP-intermediary ש-HTTP-content עובר דרכו לצורך הוספת הדברים הבאים למסמך:
  - a. הגדרות של האלמנטים של המסמך, כולם או חלקם, כ-editable, או לחילופין - לחצן שלחיצה עליו מפעילה callback המבוסס על client-side-language ושעובר על האלמנטים של המסמך ומשנה את הגדרתם ל-editable
  - b. קטע client-side-active-code שמופעל מיד עם טעינת המסמך בדפדפן או עם הלחיצה על הלחצן לעיל, שמבצע:
    - i. מעבר על כל עץ ה-HTML של המסמך והעתקתו למשתנים אחרים
    - ii. אופציונלית, החלפת הלחצן הנ"ל בלחצן לשמירת השינויים, או הוספת לחצן כזה. הלחצן יבצע את השמירה ע"י callback שיתואר ב-1c
  - c. Callback שיופעל ע"י הלחצן לעין ו/או trigger כלשהו אחר, שיבצע את הפעולות הבאות:
    - i. מעבר על כל עץ ה-HTML של המסמך והשוואתו לעץ שנשמר בסעיף 1bi
    - ii. איתור הבדלים בין העצים וע"י כך הבנה מה השינויים המדויקים שהמשתמש ביצע
    - iii. עיבוד השינוי ו/או שמירתו, כך שבעתיד, ה-HTTP-intermediary יבצע אותו על תכנים העוברים דרכו
    - iv. העברת המסמך למצב שיאפשר המשך עבודה, או המשך עריכה, אן המשך גלישה.
2. שיטה כמו claim 1, כאשר ה-HTTP-intermediary הוא HTTP-server שנמצא בין האתר או התוכנה לבין הדפדפנים.
3. שיטה כמו claim 1, כאשר ה-HTTP-intermediary הוא HTTP-proxy שנמצא בין ה-HTTP server לבין הדפדפנים.
4. שיטה כמו claim 1, כאשר התוכנית מבוססת ה-client-side-language כתובה ב-JavaScript.
5. שיטה כמו claim 1, כאשר העיבוד של השינוי (סעיף c) הוא שיגורו ל-HTTP-intermediary.
6. שיטה כמו claim 1, כאשר העיבוד של השינוי (סעיף c) הוא פתיחת dialog-box שיאפשר להוסיף ו/או להחליף הגדרות לגבי אופי השינוי המדויק, ולבצע בתום השינוי submit שישגר את השינוי הסופי אל ה-HTTP-intermediary.
7. שיטה כמו claim 6, כאשר הטקסט המקורי שאותו יש להחליף בטקסט חדש, מוצג ב-input-field שהמשתמש יכול לשנותו.
8. שיטה כמו claim 6, כאשר המשתמש יכול לציין שהטקסט המקורי לא חייב להופיע במדויק כדי להיות מוחלף, אלא מספיק approximate match (כמו למשל ב-regular-expression).
9. שיטה כמו claim 6, כאשר הטקסט החדש המחליף את הטקסט המקורי מוצג ב-input-field שהמשתמש יכול לשנותו.
10. שיטה כמו claim 9, כאשר במקום להקליד טקסט חדש, המשתמש יכול לבחור מילה מתוך אוסף טקסטים המוצג לו.
11. שיטה כמו claim 10, כאשר אוסף הטקסטים נלקח מהתרגומים השונים של הטקסט המקורי משפת מקור נתונה לשפת יעד נתונה.
12. שיטה כמו claim 10, כאשר אוסף הטקסטים נלקח מהטקסטים הקודמים שהמשתמשים השונים הציבו במקום טקסטים מקוריים הדומים לטקסט המקורי הנוכחי (Translation Memory).

13. שיטה כמו claim 11 או claim 12, כאשר התרגומים לטקסט המקורי נלקחים מה-intermediary באופן שקוף למשתמש, בטכניקה הבאה:  
 a. בנוסף ל-dialog-box, נוצר frame נוסף נסתר (size=0, border=0)  
 b. ה-hidden-frame משגר אוטומטית את הטקסט המקורי ל-intermediary ע"י HTTP request  
 c. ה-intermediary משגר כ-response דף שמוגדר שמיד עם טעינתו בדפדפן הוא יגש לשדה ב-dialog-box בו מוצגים התרגומים השונים למשתמש (בדרך כלל selection-box), או לשדה בו מוצגות אופציות ה-Translation-Memory השונות, ויציב בשדה זה את הערכים המתאימים שהתקבלו מה-intermediary.
14. שיטה כמו claim 6, כאשר ניתן לציין (באמצעות כתובת או שם קובץ) את הדף המסוים שעליו יש לבצע את ה-transformation.
15. שיטה כמו claim 14, כאשר בשדה הקלט בו המשתמש מציין את הדף המסוים, מוזן אוטומטית ערך התחלתי המתייחס לדף המסוים שבו ביצע המשתמש את השינויים.
16. שיטה כמו claim 14, אך עם תוספת של check-box שבאמצעותו ניתן לציין שהשינויים יתייחס לא לדף מסוים אלא לכל דף שהוא.
17. שיטה כמו claim 14, כאשר המשתמש יכול לציין את הדף המסוים לאו דווקא במדויק אלא באופן שיספיק approximate match (כמו למשל ב-regular-expression).
18. שיטה כמו claim 6, עם שדה/ות קלט המציין/ים את ערך/י ה-HTTP-cookies שחייב/ים להיות מוגדר/ים כדי שהשינוי יתבצע.
19. שיטה כמו claim 18, כאשר בשדה/ות הקלט הנ"ל כבר מוזן אוטומטית ערך התחלתי על פי הערך ששוגר עבור הדף אותו שינה המשתמש.
20. שיטה כמו claim 6, עם שדה/ות קלט המציין/ים את ערך/י ה-HTTP-parameters שחייב/ים להיות מוגדר/ים כדי שהשינוי יתבצע.
21. שיטה כמו claim 20, כאשר בשדה/ות הקלט הנ"ל כבר מוזן אוטומטית ערך התחלתי על פי הערך ששוגר עבור הדף אותו שינה המשתמש.
22. שיטה כמו claim 6, עם שדה/ות קלט המציין/ים את ערך/י ה-HTTP-headers שחייב/ים להיות מוגדר/ים כדי שהשינוי יתבצע.
23. שיטה כמו claim 22, כאשר בשדה/ות הקלט הנ"ל כבר מוזן אוטומטית ערך התחלתי על פי הערך ששוגר עבור הדף אותו שינה המשתמש.
24. שיטה כמו claim 6, עם שדה קלט המציין את שם המשתמש שרק עבורו השינוי יתבצע או לא.
25. שיטה כמו claim 24, כאשר בשדה/ות הקלט הנ"ל כבר מוזן אוטומטית ערך התחלתי על פי הערך ששוגר עבור הדף אותו שינה המשתמש.
26. שיטה כמו claim 6, עם שדה קלט שבו ניתן להזין XPath שיציין את ה-context שרק בקטעי מסמך העומדים בקריטריון זה יבוצע השינוי.

27. שיטה כמו claim 26 כאשר בשדה ה-XPath כבר מוזן אוטומטית ערך התחלתי המייצג XPath מינימלי המתאר את ה-context בו ביצע המשתמש את השינויים בדף.
28. שיטה כמו claim 26 כאשר בשדה ה-XPath כבר מוזן אוטומטית ערך התחלתי המייצג XPath מפורט וספציפי המתאר את ה-context המדויק בו ביצע המשתמש את השינויים בדף.
29. שיטה כמו claim 26 עם לחצן שלחיצה עליו מפעילה callback, שעובר על החלון שבו מוצג המסמך אותו משנים, ומסמן את כל הקטעים העומדים בקריטריון המצויין על ידי ה-XPath. הסימון יכול להתבצע ע"י הבהוב, צביעה בצבע שונה, הדגשה, וכו'.
30. שיטה כמו claim 6, עם לחצן שמפעיל callback שמבצע Preview ו/או Unpreview של המסמך, כלומר מציג את המסמך לפני ו/או אחרי השינוי.
31. שימוש ב-HTTP-intermediary ש-HTTP-content עובר דרכו כדי לאפשר למשתמשי דפדפנים לשנות אלמנטים ב-HTTP-content הנ"ל תוך שימוש בתוכנות עריכה המופעלות ע"י ה-HTTP intermediary ב-remote, כל זאת בשיטה הבאה:
- במידה והמשתמש לא נמצא במצב עריכה, מתן אפשרות למשתמש לעבור למצב זה ע"י לחיצה על לחצן שה-HTTP-intermediary הוסיף ל-HTTP-content, או trigger כלשהו אחר
  - מתן אפשרות למשתמש לסמן את האלמנט שברצונו לערוך (למשל ע"י לחיצה)
  - הדפדפן מדווח ל-HTTP-intermediary על פעולת המשתמש. הדפדפן עושה זאת בעקבות אחד מהדברים הבאים שהוצמד לאלמנט:
    - hyperlink
    - callback
    - watch שעוקב אחרי שינויים או פעולות שמתבצעות על האלמנט
  - ה-HTTP-intermediary פותח display וירטואלי בצד שלו, שניתן "להקרינו" למחשבים אחרים ע"י remote-display
  - אם תוכנת העריכה הנ"ל פועלת רק על קבצים, והאלמנט הנ"ל אינו קובץ בדיסק של המחשב בו מופעלת תוכנת העריכה, אזי ה-HTTP-intermediary שומר את האלמנט כקובץ בדיסק
  - ה-HTTP-intermediary מפעיל את תוכנת העריכה, ומעביר אליה כפרמטר את האלמנט, או את שם הקובץ המכיל אותו
  - ה-HTTP-intermediary משגר אל הדפדפן, בתור response ל-request שהופעל בסעיף c, client-side-active-code, המשמש כ-viewer המסוגל לתפקד כ-client של ה-remote display שהופעל בסעיף d
  - מתן אפשרות למשתמש לערוך את האלמנט ולסיים את עבודתו
  - במידה והמשתמש ביצע save במהלך עבודתו או בסיומה, ה-HTTP-intermediary מעביר את התוצאה שנשמרה למקום ו/או למחשב בו היא אמורה להיות כדי שתוצג בעתיד כשדפדפנים מבקשים להציג את האלמנט הנ"ל.
32. שיטה כמו claim 31, כאשר האלמנט הוא גרפי.
33. שיטה כמו claim 31 כאשר ה-remote-display הוא בפרוטוקול VNC.
34. שיטה כמו claim 31 כאשר ה-remote-display הוא בפרוטוקול Remote Desktop Protocol (RDP).
35. שיטה כמו claim 31 כאשר תוכנת העריכה מופעלת באותו המחשב שה-HTTP-intermediary רץ עליו.

36. שיטה כמו claim 31 כאשר תוכנת העריכה מופעלת על מחשב שונה מהמחשב שה-HTTP intermediary רץ עליו. ה-HTTP-intermediary מדבר עם תוכנת העריכה באמצעות תקשורת. התוכנה משרתת את המשתמש באותו אופן כמו ב-claim-31, כלומר באמצעות remote-display.

37. שיטה כמו claim 31 כאשר ה-viewer המועבר לדפדפן הוא Java applet.

38. שיטה כמו claim 31 כאשר תוכנת העריכה עברה שינויים כדי להתאים אותה למשימה הנדרשת ו/או כדי למנוע מהמשתמש לבצע פעולות שיזיקו או שייחשבו כפירצת אבטחה. דוגמא לשינויים הנ"ל: נטרול של פעולות open/save-as, המאפשרות למשתמש לנווט בתוכן הדיסק של המחשב של ה-HTTP-intermediary, או אף לשנות בו קבצים.

39. שיטה כמו claim 31 כאשר ה-window-manager של ה-display הוירטואלי מנוטרל, או אפשרויות מסוימות שלו מנוטרלות, כדי להתאים אותו למשימה הנדרשת ו/או כדי למנוע מהמשתמש לבצע פעולות שיזיקו או שייחשבו כפירצת אבטחה.

For the applicant,



Fenster & Co. Intellectual Property 2002, Ltd.  
c: 018/03747

Figure 1

DHL Tracking - Microsoft Internet Explorer		http://netmask.it/netmask_admin/	
Original Text:			
Text	is	Status	
<input type="checkbox"/> Min XPath:	<input type="checkbox"/> Case Sensitive	<input type="checkbox"/> Mark Current XPath	
/TABLE/TBODY/TR/TD/TABLE/TBODY/TR/TD/FONT/B			
<input type="checkbox"/> Max XPath:	/TABLE/TBODY/TR/TD/TABLE/TBODY/TR/TD/FONT[@size="2"]		
New Text:			
From Dictionary		From Translation Memory	(leave text as is)
Page:			
URL:	RegEx	http://www.dhl.com/cgi-bin/tracking.pl?TID=GB ENG&AWB=559730	
Cookie:	any		
OK		Cancel	

DHL Tracking - Microsoft Internet Explorer		http://www.dhl.com/cgi-bin/tracking.pl?TID=GB ENG&AWB=559730	
Save Replace Cancel Preview Logout			
<p>Signed for by: ALON                  ✓ Shipment delivered July 06, 2003 15:40                  Signed for by: ORLY                  Shipment delivered August 18, 2003 13:48 ✓</p>			

**This Page is Inserted by IFW Indexing and Scanning  
Operations and is not part of the Official Record**

**BEST AVAILABLE IMAGES**

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images include but are not limited to the items checked:

- ☐ BLACK BORDERS
- ☐ IMAGE CUT OFF AT TOP, BOTTOM OR SIDES
- ☐ FADED TEXT OR DRAWING
- ☒ BLURRED OR ILLEGIBLE TEXT OR DRAWING
- ☐ SKEWED/SLANTED IMAGES
- ☐ COLOR OR BLACK AND WHITE PHOTOGRAPHS
- ☐ GRAY SCALE DOCUMENTS
- ☒ LINES OR MARKS ON ORIGINAL DOCUMENT
- ☐ REFERENCE(S) OR EXHIBIT(S) SUBMITTED ARE POOR QUALITY
- ☐ OTHER: \_\_\_\_\_

**IMAGES ARE BEST AVAILABLE COPY.**

**As rescanning these documents will not correct the image problems checked, please do not report these problems to the IFW Image Problem Mailbox.**